REMARKS

Reconsideration of the above-referenced application is respectively requested in view of these remarks. Claims 1-15 are currently pending.

Claims 1, 4-6, and 9-15 under 35 U.S.C. § 103(a) as being unpatentable over Holte-Rost et al., and claims 2, 3, 7 and 8 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Holte-Rost in view of Metz.

Applicants' invention relates to a method and apparatus for stabilizing calls during a system upgrade or downgrade. The method generally involves the use of a control block which contains the version number of the application operating on both a primary and secondary controller. The primary controller writes state data to its control block, a checkpointing service replicates the data to the control block of the secondary controller, wherein the secondary controller is capable of reading the saved state data to assume processing control if necessary. If the secondary controller assumes system control and is operating on a different application version, the control block may coordinate appropriate version format conversions. The method of the present invention may be implemented, for example, by a stabilization system. The system may include a primary controller and secondary controller. Each of the primary and secondary controllers is coupled via a checkpointing service.

During normal operations, the primary controller and the secondary controller are configured to operate the same version of the application software. During such operations the primary controller writes stable and transient data to its local database. When the primary controller reaches a steady state the stable data is written to the replica state database within the control block. The checkpoint service is notified that the state data is available for transfer to the secondary controller and replicates the state data and stores it in the replica state database. In the event of a fault or failure in the primary controller, the system shuts down the primary controller to ensure the controller has the opportunity to update the replica state database and, via the checkpoint service, the replica state database. Upon shutdown of the primary controller, the secondary controller assumes processing control of the system. The secondary controller reads the replica state database, rebuilds its local database, and is therefore able to take control with little or no interruption of wireless service. Thus, and as required by the claims, the

converted state data ensures stability so that ongoing operations, including calls in a wireless communication system, are maintained.

An upgrade or downgrade of service generally entails the installation of new application software or hardware on the secondary controller. After the secondary controller has its software upgraded or downgraded, the secondary controller prepares to assume control of the system and notifies the secondary control block version table of the new application version number. Then, the checkpoint service communicates with the primary control block and updates the control block version table to indicate the new secondary application version. The primary controller then shuts down or quiesces. With the secondary controller ready to assume control of the system after the shut down of the primary controller, the primary processor compares versions of the primary application and secondary application to determine if the change in the secondary application was an upgrade, a downgrade or no change. If the new version is a downgrade, the saved state data is converted to a format compatible to the older application version running, the converted data is rewritten to the replica state database, the checkpointing service is notified and the replica state database is updated. If it is determined that the new version is an upgrade or no change, conversion of the state data may be delayed. The secondary controller takes over primary control of the system and opens the checkpoint replica database for read and write access. Next, the system determines if the replica state data needs to be converted, when the new application is an upgrade, and the replica state data is converted to the new version format.

According the invention, operations, such as wireless calls, that are running on the primary controller operate on the secondary controller without interruption regardless of whether the software is being upgraded or downgraded. The system operates without interruption because the state data is accessible to both the first and secondary controller.

The Office Action rejects claims 1, 4-6, and 9-15 under 35 U.S.C. § 103(a) as being unpatentable over U.S. patent No. 6,101,327 to Holte-Rost et al. It is stated that Holte-Rost disclose the elements of independent claims 1, 6, 11, 14 and 15 except that they do not specifically disclose the "first format" and the "second format" but that it would be obvious to one of ordinary skill in the art to understand that the disclosed resource object was storing a state in a "first format" and after the transfer of the resource

object the state was held in a "second format." In addition, it is stated that it would be obvious in view of Holte-Rost to convert the saved state data in a first format of the state data to a second format of the state data, wherein the second format of the state data is compatible with the upgraded second version of a control application.

With respect to claims 1, 4-5 and 11-14, Applicants claim that the upgraded second version of software runs on the secondary controller when the secondary controller has operational control of the system and after the primary processor has quiesced. In this way, only one version of software is running at time to operate the system. On the other hand, Holte-Rost teaches that the old release and the new release of software are active and running at the same time such that both the old release and the new release have operational control of the system at the same time. In the wireless communication system, this means that the primary controller may operate the old software, and therefore the system, for a set of calls while the secondary controller may operate the new software, and therefore the system, for another set of calls. Thus both processors and software versions have operational control of the system simultaneously and contrary to the present claims. See column 6, lines 8-12. This is directly contrary to the present invention as found in claims 1, 4-5 and 11-14.

It is stated in the Office Action in response to Applicants' argument about the upgraded second version of software runs after the primary controller has quiesced is not persuasive because the primary controller attempts to "gracefully shutdown." Applicant has refined the claim so that the issue is no longer that the controllers are operating but which controller has operational control of the system. The Examiner interprets this language from the specification to mean that the second controller begins a continuation of the execution while the first controller is still operating. Applicant respectfully traverses this argument. According to the language of the claims, the second controller does not have operational control of the system with the upgraded version of the control application until after the quiescing of the primary controller. In other words, quiescence is completed when the second controller begins to control the operation of the system. This is supported by the Specification when it is stated that "Upon shutdown of the primary controller 52, the secondary controller 54 assumes processing control of the system 50." See page 9, lines 1-2.

With the language in the claims focusing on the operational control of the system, the previous comments made in the Office Action regarding the CommitTakeover still do not apply. During the CommitTakeover process described by Holte-Rost both the new and old software is running at the same time and both versions of software are controlling the operation of the system depending on which call that is being handled. Holte-Rost is replete with examples of when the new and old software versions are running at the same time, especially during testing of the new software. On the other hand, the present invention of claims 1, 11 and 14 are directed so that two versions are not running at the same time and in particular where one controller running one version of software has operational control of the system.

To continue, the Advisory Action states that the second controller is running in the background and the second controller operates after the primary controller is quiesced. As stated, Applicants have amended to the claims to further clarify that the second controller assumes operational control of the system after the quiescence of the primary controller. Thus, there is only one controller that is controlling the system at any given time even though both the primary and secondary controller may be functional at the same time. As seen in the claims, the issue is operational control of the entire system not the individual operation of each of the controllers.

With respect to claims to claims 6 and 15, Applicants continues to respectfully traverse the interpretation of the present claims and the scope of Holte-Rost. Claims 6 and 15 are directed to a downgrading a software version of a wireless communication system or a broader system. In the present invention, the primary controller controls the operation of the system and the second controller is used for backup in the event that something prevents the first controller from operating. One of the elements of concern in transfer from the first controller to the second controller is the transfer of state data, which indicates the steady state, from the first controller to the second controller. This concern is augmented in the context of software downgrades when the software versions on the first controller and the second controller are not the same and where software is not downwardly compatible, i.e. the state data from a new version of software cannot run on an old version of software. In order to overcome this concern, claims 6 and 15 each include a converting step that converts the state data saved in a first format corresponding

to the state data for running the first version of software to state data in a second format corresponding the state data needed for running the second version of software. As state in the Specification for a system downgrade, the conversion of the state data is to an older version of the state data. See Page 10, lines 14-18. The conversion of state data to be compatible to the version of software allows the present invention to ensure the stability of the system, including maintaining the operation of ongoing calls.

To ensure the stability of the system and ongoing calls, claims 6 and 15 also indicate that the secondary controller assumes operational control of the system after the quiescence of the primary controller. Thus, only one processor is operating at a time and the secondary controller operates with the converted software. Holte-Rost discloses that the primary controller and the secondary controller have operational control of the system at the same time during testing. Moreover, the secondary controller only assumes complete operational control of the system when the all of the old processes on the primary controller are complete.

Holte-Rost does not disclose the conversion of the state data for a downgrade. Instead, Holte-Rost discloses only that resources are saved for use by the new version and the old version where the new version is an upgrade. In the Advisory Action, it is stated that a conversion of data for an upgrade is the same as conversion of data for a downgrade. Holte-Rost does not support this. The new version discussed in Holte-Rost is for new *upgraded* software. Nowhere in Holte-Rost is there a discussion of how to downgrade the software to an earlier version of the software. Holte-Rost teaches away from a conversion when it discusses the cancellation, or abortion, of a software conversion. Holte-Rost states, "A reversion, by applying the signal CommitTakeover to the old static processes, is possible but the states in the new static processes that have been changed during the time between the Takeover signal and the CommitTakeover signal will be lost." Holte-Rost, column 8, lines 58-63. In other words, the state data transferred by Holte-Rost in a downgrade is the state data from the new version of software that cannot be used by the downgraded second version. Thus, the new versions that are downwardly compatible will be discarded. If the state data could be used by the downgraded second version then signals would not be lost. This does not create network stability as is required by claims 6 and 15.

The Examiner admits that the some of the data from a new version of software will not be acceptable to a downgraded version of the software and that is the reason that the data is converted. As stated above, Holte-Rost does not state that the downgraded state data is converted. In fact, the data for downgrades is discarded because it is not compatible. This is also supported by extrinsic evidence. As explained in the enclosed excerpt from *Release Notes for Cisco CallManager Release 5.0(1),* (2006), "You can also back out of an upgrade by restarting the system using the software version on the inactive partition. However, if you have made any configuration changes since you installed the upgrade, they get lost when you revert to the older version of the software." *Release Notes for Cisco CallManager Release 5.0(1),* http://www.cisco.com/en/US/products/sw/voicesw/ps556/prod_release_note09186a00805e7e74.html (2006) page 5. (Due to the length of the document, Applicants only include the first 10 pages.) By loosing configuration changes, the stability of the system is not ensured as operation of certain items is no longer possible. As stated in the claims, the conversion of the state data to be compatible with the downgraded version of the software permits the stability of the system.

Applicants therefore disagree that downgrading software from a one version to an earlier version is equivalent to upgrading. In particular, the primary concern is that software is not always backward compatible so that not all the data, such as state data, used and created with the new software does not necessarily run on the earlier version of software. The present invention overcomes this issue by converting the state data. Holte-Rost does not overcome this problem because it does not convert saved state data.

With the amendment to clarify that only one controller has operational control of the system, the importance of the converting step is more evident. The converting step, which is present in both the upgrading and downgrading functions, takes to the state data for what is operating on one processor and therefore one version of software and coverts to state data to a format compatible with the second version of software running on the second processor. The conversion of the state data is essential because only one controller, and therefore one version of software, is controlling the operation of the system. If the conversion of state data is not performed then the functions of the system, e.g. ongoing calls, are no longer operational and will fail.

Holte-Rost does not have this issue. As stated in the Summary, "A smooth change which transfers states to a new version is accomplished by allowing ongoing transactions, i.e., 'old traffic' to run to completion using the old software version. The processes containing the states which are to be transferred from the old to the new version of software are, at the beginning of the software change, in the control of the old software. By means of different synchronization signals the new software will be able to access the states of the processes in control of the old software on an 'as needed basis', so as to finally become the owner of the processes containing the updated states . . . ." Column 3, lines 41-54. The present invention cannot allow old traffic to complete using the old version of software while the new traffic uses the new version of software because only one version of software has operational control of the system at any given time. Thus, the present invention does not include any synchronization signals to access the states of the old software on any basis because only one state is used.

Holte-Rost describes a transfer of states in connection with Fig. 6. See column 9, line 22 to column 10, line 6. In this description, Holte-Rost describes various different phases of the upgrading process. At points during the upgrading process, the old version is running while the new version is being tested. This conversion process does not disclose, teach or suggest how to convert state data to a form that is compatible with the new version of software when there is only one controller having operational control.

In view of the foregoing, it is respectfully submitted that Holte-Rost does not disclose, teach or otherwise suggest the invention claimed in independent claims 1, 6, 11, 14 and 15. As claims 4-5, 9-10 and 12-13 depend upon and include each and every limitation of claims 1, 6 and 11, respectively, it is respectfully submitted that Holte-Rost also do not render the dependent claims obvious. It is therefore respectfully requested that the rejection under Section 103(a) be withdrawn.

In the Office Action, claims 2, 3, 7 and 8 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Holte-Rost in view of United States Patent No. 5,666,293 to Metz et al. Assuming that Metz does disclose the use of a control table including software version, Metz does not disclose the quiescing of primary processor before running the updated second version of software as required by claims 2 and 3. In addition, Metz does not disclose the converting of state data as required by claims 7 and

8. For the reasons given above with respect to independent claims 1 and 6 upon which claims 2, 3, 7 and 8 depend, Applicants respectfully submit that these rejected claims are patentable over Holte-Rost in view of Metz. Applicant therefore respectfully request that this rejection under Section 103(a) be withdrawn.

As the Applicant has overcome all substantive rejections and objections given by the Examiner and have complied with all requests properly presented by the Examiner, the Applicant contends that this Amendment, with the above discussion, overcomes the Examiner's objections to and rejections of the pending claims. Therefore, the Applicant respectfully solicits allowance of the application. If the Examiner is of the opinion that any issues regarding the status of the claims remain after this response, the Examiner is invited to contact the undersigned representative to expedite resolution of the matter.

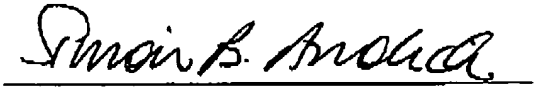
Please charge any fees associated herewith, including extension of time fees, to **50-2117.**

Respectfully submitted,
Moore, Brian J.

SEND CORRESPONDENCE TO:

Motorola, Inc.
Law Department

Customer Number: **22917**

By: 

Simon B. Anolick
Attorney for Applicant
Registration No.: 37,585
Telephone: 847-576-4234
Fax: 847-576-3750

Release Notes for Cisco CallManager Release 5.0(1)

Feb. 28, 2006

These release notes describe the new features and caveats for Cisco CallManager release 5.0(1).

To view the release notes for previous versions of Cisco CallManager, choose the Cisco CallManager version from the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/voice/c_callmg/index.htm

Before you install Cisco CallManager, Cisco recommends that you review the "Important Notes" section on page 109 for information about issues that may affect your system.

Note Cisco recommends that you check Cisco.com for the latest software updates to Cisco CallManager and its applications, and download and install the latest updates on your system before the deployment of your Cisco CallManager system. For a list of commonly used URLs, see the "Upgrading System Software" section on page 3.

Contents

These release notes discuss the following topics:

- Introduction, page 2
- System Requirements, page 2
- Related Documentation, page 4
- New and Changed Information, page 4
- Installation Notes, page 109
- Limitations and Restrictions, page 109
- Important Notes, page 109
- Caveats, page 113
- Troubleshooting, page 130

Cisco Systems

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

- Documentation Updates, page 130
- Obtaining Documentation, page 137
- Documentation Feedback, page 138
- Cisco Product Security Overview, page 138
- Obtaining Technical Assistance, page 139
- Obtaining Additional Publications and Information, page 141

Introduction

Cisco CallManager, a network business communication system, provides high-quality telephony over IP networks. Cisco CallManager enables the conversion of conventional, proprietary, circuit-switched PBXs to multiservice, open LAN systems.

System Requirements

Make sure that you install and configure Cisco CallManager release 5.0(1) on a Cisco Media Convergence Server (MCS).

You may also install Cisco CallManager on a Cisco-approved HP server configuration or a Cisco-approved IBM server configuration.

Cisco CallManager 5.0 requires a minimum of the following on the Cisco CallManager servers.

- 2 GB of memory
- 72 GB Disk Drive
- 2 GHz processor

Cisco recommends that you connect each Cisco CallManager node to an Uninterruptible Power Supply (UPS) to provide backup power and protect your system against a power failure.

Supported Platforms

Cisco CallManager 5.0 supports the following list of servers without modification:

- MCS-7815I-I1-IPC3
- MCS-7825-I1-IPC1
- MCS-7825-H1-IPC1
- MCS-7835-I1-IPC1
- MCS-7835-H1-IPC1
- MCS-7845-I1-IPC1
- MCS-7845-H1-IPC1
- MCS-7845H-2.4-EVV1
- MCS-7845H-3.0-IPC1

Cisco CallManager 5.0 supports the following list of servers when additional memory is added.

- MCS-7815I-3000
- MCS-7815I-[illegible]-IPC1
- MCS-7825I-3000
- MCS-7825I-3.0-IPC1

Cisco CallManager 5.0 supports the following list of servers when the disk drives are replaced with a 72 GB or 80 GB hard drive and additional memory is added to meet the minimum memory requirement.

- MCS-7815I-2000
- MCS-7825H-2266
- MCS-7825H-3000
- MCS-7825H-2.2-EVV1
- MCS-7835H-2.4-EVV1
- MCS-7835I-2.4-EVV1
- MCS-7835H-3.0-IPC1

Determining the Software Version

To determine the software version of Cisco CallManager, open Cisco CallManager Administration. The following information displays:

- Cisco CallManager System version
- Cisco CallManager Administration version

Upgrading System Software

You can access the latest software upgrades for Cisco CallManager 5.0 on Cisco.com. Table 1 lists the URLs where you access the software.

Table 1

Software	Download URL
Cisco CallManager 5.0	http://www.cisco.com/cgi-bin/tablebuild.pl/callmgr-50
Locale installers	http://www.cisco.com/kobayashi/sw-center/telephony/callmgr/locale-installer.shtm
Phone firmware	http://www.cisco.com/cgi-bin/tablebuild.pl/ip-7900ser http://www.cisco.com/cgi-bin/tablebuild.pl/ip-7900ser-crypto
Cisco Security Agent (CSA)	http://www.cisco.com/cgi-bin/tablebuild.pl/cmva-3des

Related Documentation

Refer to the Cisco CallManager Document Guide for a list of documents that are related to Cisco CallManager release 5.0 at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/voice/c_callmg/5_0/doc_gd/index.htm

New and Changed Information

The following sections describe new features and changes that are pertinent to this release of Cisco CallManager. The sections may include configuration tips for the administrator, information about users, and where to find more information.

- Installation, Upgrade, Migration, and Disaster Recovery, page 4
- Cisco IP Telephony Platform Administration, page 7
- Cisco CallManager Administration, page 9
- Cisco CallManager Features, page 19
- Cisco CallManager User Options Web Pages, page 56
- Cisco CallManager Applications, page 57
- Bulk Administration Tool Features, page 73
- TAPS Features, page 76
- Security Features, page 78
- New and Changed Information for Cisco CallManager Serviceability, page 83
- New and Changed Information for Cisco IP Phones, page 90
- New and Changed Information for Third-Party API, page 101

Installation, Upgrade, Migration, and Disaster Recovery

This following sections describe the changes made to the installation, upgrade, and disaster recovery procedures in Cisco CallManager 5.0(1):

- Installation Overview, page 4
- Software Upgrades, page 5
- Data Migration Assistant (DMA), page 5
- Disaster Recovery Enhancements, page 6
- Where to Find More Information, page 7

Installation Overview

Cisco CallManager 5.0(1) uses a different installation framework than previous releases. The installation process allows you to perform a basic installation, upgrade from Cisco CallManager 4.x to Cisco CallManager 5.0, and upgrade to a newer service release during the installation.

Note Although you do not need a license to install Cisco CallManager 5.0(1), you must have a Cisco CallManager server license to activate services on the server and you must have phone licenses to add phones to the Cisco CallManager database.

For a more detailed description of the different installation types, see Table 2.

Table 2 Installation Options

Installation Types	Description
Basic Install	This option represents the basic Cisco CallManager 5.0(1) installation, which installs the software from the installation disc and does not use any imported data.
Upgrade During Install	This option allows you to upgrade the software version contained on the installation disc with the latest service release. You can also choose Upgrade During Install followed by a Windows Upgrade and perform both during the installation process.
Windows Upgrade	This option allows you to import database information from a Cisco CallManager 4.x system using a file produced by the Data Migration Assistant (DMA) tool.

Software Upgrades

With Cisco CallManager 5.0, you can install software upgrades on your server while the system continues to operate. Multiple partitions exist on your system disk, including an active, bootable partition and an inactive, bootable partition. The system boots up on the partition that is marked as the active partition.

When you install the software upgrades by using the Cisco IP Telephony Platform interface, you install the software on the inactive partition. The system continues to function normally while you are installing the software. When you are ready, you activate the inactive partition and reboot the system with the new upgrade software. The current active partition will then get identified as the inactive partition when the system restarts. The current software remains in the inactive partition until the next upgrade.

You can also back out of an upgrade by restarting the system using the software version on the inactive partition. However, if you have made any configuration changes since you installed the upgrade, they get lost when you revert to the older version of the software.

Data Migration Assistant (DMA)

The Cisco DMA assists you with the first step in migrating Cisco CallManager 4.x data to Cisco CallManager 5.0 by backing up Cisco CallManager 4.x data in a format that Cisco CallManager 5.0 can read. Cisco CallManager 4.x runs in a Windows environment, and Cisco CallManager 5.0 runs in a Linux environment, so the Cisco DMA exports Windows-based data to a format that Cisco CallManager 5.0 can import. The Cisco CallManager 5.0 installation process converts the backed up data as needed for Cisco CallManager 5.0, which completes the data migration.

The Cisco DMA saves the data that it exports in a tape archive (tar) file in a location that you specify.

You must install and run the Cisco DMA on the Cisco CallManager publisher server before you upgrade to Cisco CallManager 5.0. If you make any Cisco CallManager configuration changes after running the the Cisco DMA, the system does not retain these changes when you upgrade.

In addition to exporting Cisco CallManager data, the Cisco DMA exports data for these related applications:

- Attendant Console (AC)
- Cisco CallManager Extension Mobility (EM). DMA does not export the last user logged in data.
- CDR Analysis and Reporting (CAR)
- Certificate Authority Proxy Function (CAPF)
- Certificate Trust List (CTL)

Note If you installed the CAPF utility 1.0(1) on a Cisco CallManager 4.0 subscriber server, you must copy the CAPF data to the 4.0 publisher database server before you upgrade to Cisco CallManager 5.0. Failing to perform this task causes a loss of CAPF data.

The Cisco DMA does not export this information:

- Custom Music on Hold (MOH) files—You must reapply these files after you upgrade to Cisco CallManager 5.0.
- TFTP phone load files—You must reapply these files after you upgrade to Cisco CallManager 5.0.
- Files on Cisco CallManager subscriber servers—Subscriber servers obtain required information from the publisher server as part of the Cisco CallManager upgrade process.
- Last Logged User in Extension Mobility—Extension mobility users will need to enter the user name and PIN to login the first time after the upgrade. The user does not need to enter the user name on subsequent logins.
- CDR database—If you want to preserve the historical data in the CDR database, you must back up the data in the CDR database. You can use the backup utility on Microsoft SQL server.

Disaster Recovery Enhancements

The Cisco Disaster Recovery System (DRS), accessed from the Cisco CallManager 5.0(1) Administration window, provides full data backup and restore capabilities for all servers in a Cisco CallManager cluster. Cisco DRS allows you to perform regularly scheduled automatic or manually initiated data backups and system restoration.

The Cisco Disaster Recovery System performs a cluster-level backup, which means that it collects backups from all servers in a Cisco CallManager cluster to a central location, then combines the backups into a single volume (or multiple volumes if necessary), and archives the backup data to a physical storage device.

When performing a system data restoration, you can choose which nodes and features in the cluster you want to restore.

The Cisco Disaster Recovery System features include:

- Graphical user interface for performing backup and restore tasks
- Command-line access to most Cisco DRS functions
- Scheduling engine to initiate tasks at user-specified times
- Archives backed up to a physical tape drive or to a remote server

Where to Find More Information

- *Cisco IP Telephony Disaster Recovery System Administration Guide*
- *Cisco IP Telephony Data Migration Assistant 2.0 User Guide*
- *Upgrading Cisco CallManager Release 5.0(1)*
- *Installing Cisco CallManager Release 5.0(1)*
- *Cisco IP Telephony Platform Administration Guide, Release 5.0(1)*

Cisco IP Telephony Platform Administration

For Cisco CallManager 5.0(1), you can perform many common system administration functions through the Cisco IP Telephony platform.

This chapter comprises the following topics:

- Overview, page 7
- Browser Requirements, page 8
- Platform Status and Configuration, page 8
- Restart Options, page 8
- Security Configuration, page 8
- Software Upgrades, page 9
- Services, page 9
- Command Line Interface, page 9

Overview

Cisco IP Telephony Platform Administration allows you to configure and manage the Cisco IP Telephony platform by doing these tasks:

- Check software and hardware status.
- Check and update IP addresses.
- Ping other network devices.
- Manage NTP servers.
- Upgrade system software and options.
- Restart the system.

The following sections describe each platform function in more detail.

Browser Requirements

You can access Cisco CallManager Administration, Cisco CallManager Serviceability, and Cisco IPT Administration by using the following browsers:

- Microsoft Internet Explorer version 6.0 or later
- Netscape Navigator version 7.1 or later

Note Cisco does not support or test other browsers, such as Mozilla Firefox.

Platform Status and Configuration

From the **Show** menu, you can check the status of various platform components, including:

- Cluster and nodes
- Hardware
- Network
- System
- Installed software and options

Settings

From the **Settings** menu, you can view and update the following platform settings:

- Ethernet—Updates the IP addresses and Dynamic Host Configuration Protocol (DHCP) settings that were entered when the application was installed.
- NTP Server settings—Configures the IP addresses of an external NTP server; add or delete an NTP server.
- SMTP settings—Configures the SMTP host that the platform will use for sending e-mail notifications.

Restart Options

From the **Restart** menu, you can choose from the following options for restarting or shutting down the system:

- Switch Versions—Switches the active and inactive disk partitions and restarts the system. You normally choose this option after the inactive partition has been updated and you want to start running a newer software version.
- Current Version—Restarts the system without switching partitions.
- Shutdown System—Stops all running software and shuts down the server.

Security Configuration

The platform security options enable you to manage security certificates and Secure Internet Protocol (IPSec). From the **Security** menu, you can choose the following security options:

- Certificate Management—Manages certificates, Certificate Trust Lists (CTL), and Certificate Signing Requests (CSR). You can display, upload, download, delete, and regenerate certificates. Through Certificate Management, you can also monitor the expiration dates of the certificates on the server.
- IPSEC Management—Displays or updates existing IPSEC policies; sets up new IPSEC policies and associations.

Software Upgrades

The software upgrade options enable you to upgrade the software version that is running on the platform or to install specific software options, including Cisco IP Telephony Locale Installers, dial plans, and TFTP server files.

From the **Install/Upgrade** menu option, you can upgrade system software from either a local disc or a remote server. The upgraded software gets installed on the inactive partition, and you can then restart the system and switch partitions, so the system starts running on the newer software version.

Note For Cisco CallManager 5.0, you must do all software installations and upgrades by using the Software Upgrades menu options. The system can upload and process only software that Cisco Systems approved. You cannot install or use third-party or Windows-based software applications that you may have been using with a previous version of Cisco CallManager with Cisco CallManager 5.0.

Services

The application provides the following platform utilities:

- Ping—Checks connectivity with other network devices.
- Remote Support—Sets up an account that Cisco support personnel can use to access the system. This account automatically expires after the number of days that you specify.

Command Line Interface

The command line interface, which you can access from the console or through a secure shell connection to the server, provides a subset of the platform functionality that is available through the platform user interface. Keep in mind that the command line interface is designed for system emergencies and not as a replacement for the user interface.

Where to Find More Information

- *Cisco IP Telephony Platform Administration Guide*

Cisco CallManager Administration

The Cisco CallManager 5.0 administration enhancements are described in the following sections:

- General Administration Enhancements, page 10
- Navigating to IP Telephony Applications Within Cisco CallManager, page 10
- Localizing Cisco CallManager Administration, page 10
- Configuring Servers, page 11
- Publisher and Subscriber Name Changes, page 11
- Media Resources, page 11
- Migration Tips, page 11
- General Changes Made to Multiple Windows, page 12
- Line and Phone Configuration Improvements, page 12

- System Menu Changes, page 13
- Call Routing Menu Changes (formerly Route Plan Menu), page 14
- Media Resources Menu, page 14
- Voice Mail Menu, page 14
- Device Menu Changes, page 14
- Application Menu Changes, page 15
- User Management Menu, page 16
- Bulk Administration Menu, page 16
- Service Parameter Changes, page 17
- Where to Find More Information, page 18

General Administration Enhancements

Cisco CallManager Administration Release 5.0 supports JSPs, STRUTS framework, and Java. The following requirements apply to Cisco CallManager Administration:

- Tomcat 5.0.2
- Microsoft Internet Explorer (IE) 6.0 or higher
- Netscape 7.1 or higher

Note This release does not support Microsoft IE 5.5 and Netscape 7.0.

Navigating to IP Telephony Applications Within Cisco CallManager

Cisco CallManager Administration includes a navigation bar in the upper, right corner of the window that takes the administrator to the following Cisco IP Telephony applications:

- Cisco CallManager Administration
- Cisco CallManager Serviceability
- Disaster Recovery System
- Platform Administration

Note The Bulk Administration Tool (BAT) appears as a menu item on the Cisco CallManager Administration menu.

Localizing Cisco CallManager Administration

Cisco CallManager Release 5.0 incorporates the following localization capabilities:

- End User Configuration windows get localized. Other configuration windows that share the End User Configuration get localized.
- To see the localization, set the browser to the language that is required. If that language locale is loaded, the configuration windows will be localized.